

逻辑与神经之间的桥

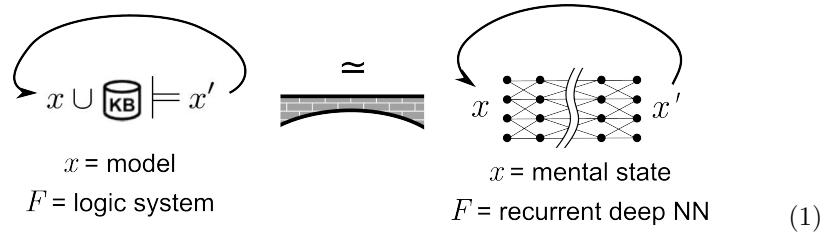
甄景贤 (King-Yin Yan)

General.Intelligence@Gmail.com

Abstract. Logic-based AI 和 connectionist AI 长久分裂，但作者最近发现可以建立两者之间的对应关系。逻辑的结构类似人类的自然语言，但大脑是用神经思考的。机器学习的主要目标，是用 inductive bias 去加快学习速度，但这目标太空泛。将逻辑结构加到神经结构之上，就增加了约束，亦即 inductive bias。

逻辑 AI 那边，「结构」很精细，但学习算法太慢；我的目的是建立一道「桥」，将逻辑 AI 的某部分结构转移到神经网络那边。

这个问题搞了很久都未能解决，因为逻辑 AI 那边的结构不是一般常见的数学结构，单是要表述出来也有很大困难。直到我应用了 model theory 的观点，才找到满意的解决方案：



首先解释 logic 那边的结构，然后再解释 neural network 那边的结构。

1 逻辑的结构

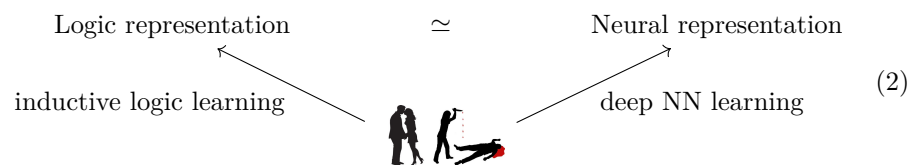
一个逻辑系统可以这样定义：

- 一些 constant symbols, predicate symbols, 和 function symbols
- 由上述的原子建立 命题 (propositions)
- 命题之间可以有连接词： \neg, \wedge, \vee 等
- 建立 逻辑后果 (consequence) 关系： $\Gamma \vdash \Delta$

个人认为 relation algebra [10] [8] 比较接近人类自然语言，但在数理逻辑研究中最通用的逻辑是 first-order logic (FOL)。然而这并不是重点，因为各种逻辑基本上是等效的，而且相互之间可以很容易地转换。以下集中讨论 FOL。

由一些原始的 sensory data，可以透过逻辑学习出一些 logic formulas，即知识库 (knowledge base) $\boxed{\text{KB}}$ 。这个过程叫逻辑诱导学习 (inductive logic programming, ILP)。学经典 AI 的人都知道 ILP，但近数十年来，注意力集中在统计学习，这种符号逻辑的学习法被忽视。

原始的 sensory data 可以透过神经网络进行模式识别，也可以透过 ILP 进行模式识别，两条路径的结果很明显应该是（近似地）isomorphic 的：



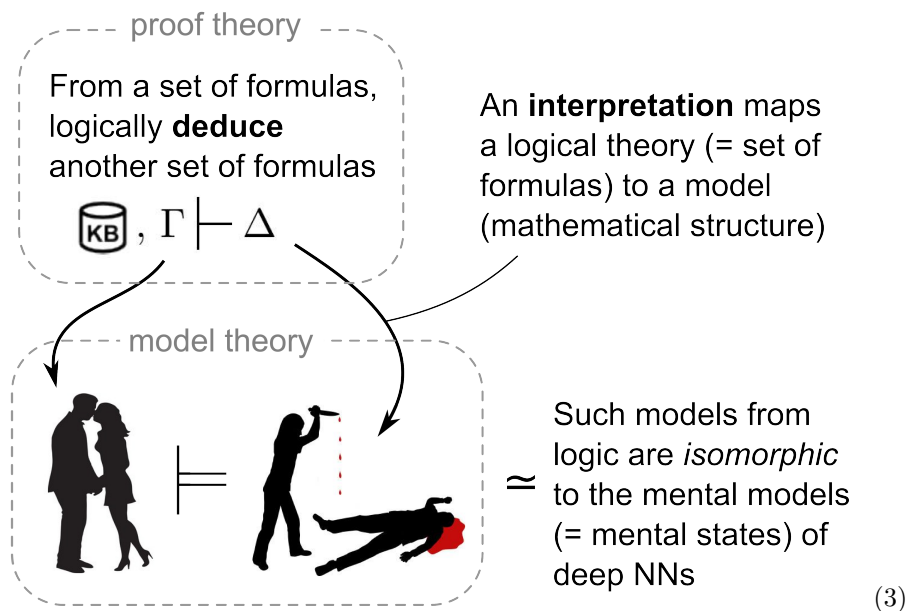
我以前花了很多时间思考怎样将逻辑的 representation 过渡到神经网络去，但发觉这个目标非常 elusive。

一方面，逻辑是几百年来发展起来的关于人类思考的规律；逻辑的描述是正确的；逻辑和神经之间必然有一个 correspondence，因为它们都在做同样的事（智能）。

在认知科学里，有很多人相信大脑的内部 representation 是一些所谓 “mental models”，而很少人会相信大脑使用一些像命题那样的符号结构做 representation，甚至用 λ -calculus 那样的符号 manipulation 去思考。

举例来说，用文字描述一起凶杀案，读者心目中会建立一个「模型」，它类似於真实经验但又不是真实的。人脑似乎是用这样的 mental models 思考，而不是一些命题的集合。

我終於发现到，logic-neuro correspondence 必须透过 model theory 才能达成：



\vdash 是指由一些（符号逻辑的）命题集合推导出新的命题集合。 \models 指的是，由一个模型推导出另一个模型必然为真。

2 模型论

模型论基础可参看 [4] [9]。模型论的做法是将逻辑的符号语言 (language \mathcal{L}) 和它所指涉的结构 \mathcal{L} -structure 分割，中间用 interpretation map 关联起来。

\mathcal{L} 就是符号的集合 (predicates, relations, functions, constants)，递归地生成出句子和复合句子。这些都是 symbolic 的东西。

\mathcal{L} -structure 可以是任何抽象代数结构，它通常包含一个 base 集合，然后在集合上定义一些函数和关系。

模型论的中心思想是透过 interpretation i 去「保存」一些关系，例如：

$$R(a, b) \xrightarrow{i} R^{\mathcal{M}}(a^{\mathcal{M}}, b^{\mathcal{M}}) \quad (4)$$

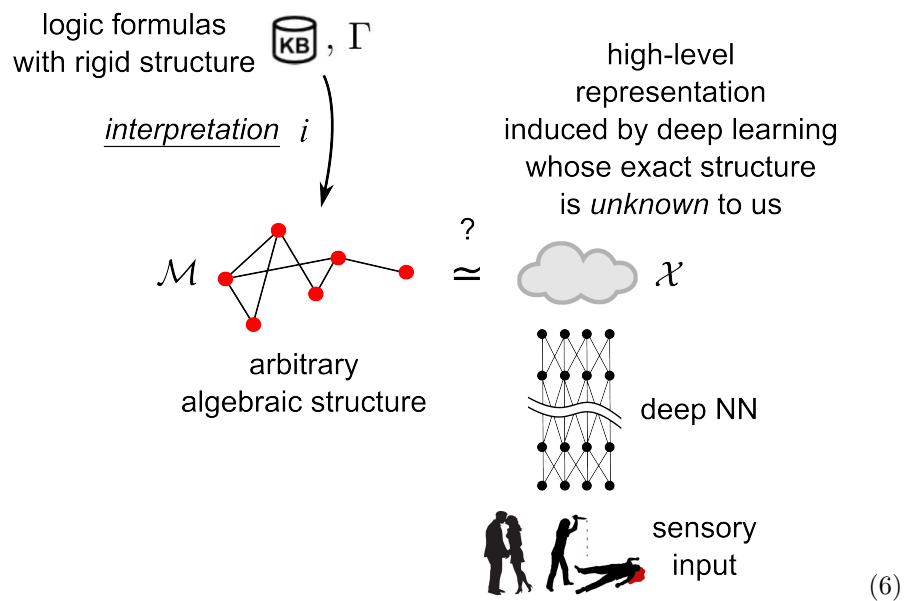
R 是一个关系， $x^{\mathcal{M}}$ 代表在结构 \mathcal{M} 之上， x 所对应的物体。左边是符号逻辑，右边是实体的结构。模型论应用在 first-order logic，得出 \vdash 和 \models 等价的结论（看起来就好像同语反覆），这在数理逻辑教科书中都有，例如 [6]。

如果用範疇论的方法表示逻辑结构和神经结构之间的对应：

$$\begin{array}{ccc} \mathcal{L} & & \\ \downarrow i & & \\ \mathcal{M} & \simeq & \mathcal{X} \\ & & \uparrow \text{deep NN} \\ & & \mathcal{S} \end{array} \quad (5)$$

- \mathcal{L} = category of logic theories (= sets of formulas)
- i = interpretation maps
- \mathcal{M} = category of models (from logic)
- \mathcal{X} = category of models (from deep NNs)
- \mathcal{S} = sensory input

上图等同於下面的卡通解释：




换句话说， $\mathcal{X} = \text{cloud}$ 是由深度学习 induce 出来的结构；但它的结构对我们来说是不透明的（这是神经网络的弱点）。

而 $\mathcal{M} = \text{graph}$ 的结构就是模型论研究的对象。

在模型论中， \mathcal{L} 是逻辑句子的範疇， $\mathcal{M} = \text{graph}$ 可以是任何抽象代数结构。只需把 \mathcal{L} 中的 constants, predicates, relations, functions 映射到 \mathcal{M} 就行。为简化

讨论，我们只考虑 constants 和 relations，因为二者是逻辑中最本质的东西。

$$\begin{array}{ccc}
 \mathcal{L} & \xrightarrow{i} & \mathcal{M} \\
 \text{constant symbol} & \mapsto & \bullet \\
 \text{relation symbol} & \mapsto & \bullet \text{---} \bullet
 \end{array} \tag{7}$$

问题是在神经那边缺乏  的结构。一直以来，人们习惯把神经网络看成是“black box”，但如果我们不知道 $\mathcal{X} = \text{cloud}$ 的结构，就无法建立 $\mathcal{M} \simeq \mathcal{X}$ 的 isomorphism。

3 神经网络的结构

那么，神经网络的 representation 究竟是什么结构？

一个神经网络基本上是：

$$F(\mathbf{x}) = \textcircled{S}(W_1 \textcircled{S}(W_2 \dots \textcircled{S}(W_L \mathbf{x}))) \tag{8}$$

其中 L 是层数， W_ℓ 是每层的权重矩阵， \textcircled{S} 是对每个分量的 sigmoid function（其作用是赋予非线性）。

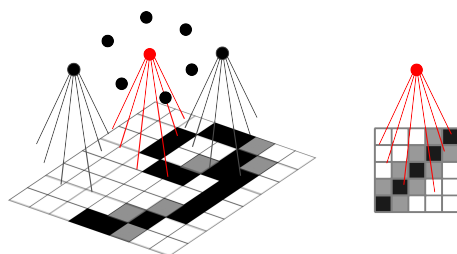
\textcircled{S} 作用在 \mathbf{x} 的每个分量上，它的作用在座标变换下没有不变性。所以 \textcircled{S} 不是一个向量运算，从而 \mathcal{X} 的结构也不是向量空间的结构。通常习惯把 \vec{x} 写成向量形式，但这是误导的。

如果将神经网络首尾相接造成迴路，这是一种智能系统的最简单形式，它的状态方程是 $\mathbf{x}_{n+1} = F(\mathbf{x}_n)$ （或连续时间的 $\dot{\mathbf{x}} = f(\mathbf{x})$ ），由此可以看出， $\mathbf{x} \in \mathcal{X}$ 是一个微分流形。更深入地讲，它是一个力学上的 Hamiltonian 系统，具有 symplectic（辛流形）结构。换句话说，它是微分流形，而且有一个辛度量 (metric)。但这超出了本文范围，详见作者的 [14]。

现在思考一下，神经网络怎样识别模式，或许会有帮助：

考虑最简单的情况，例如提取 digit “9” 的特徵的一层网络。这层网络可以有很多神经元（左图），每个神经元局部地覆盖输入层，即所谓视觉神经元的 local

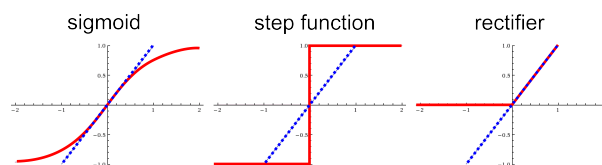
receptive field (右图)。



(9)

假设红色的神经元专门负责辨识「对角线」这一特徵。它的方程式是 $y = \sigma(Wx)$ 。矩阵 W 的作用是 affine「旋转」特徵空间，令我们想要的特徵指向某一方向。然后再用 σ 「挤压」想要的特徵和不想要的特徵。Sigmoid 之后的输出，代表某类特徵的存在与否，即 $\{0, 1\}$ 。这是一种资讯的压缩。

讲一点 chaos theory: σ^{-1} 的作用是「扯」(stretch)，将本来邻近的两点的距离非线性地拉远。看看以下各种常见的激活函数，它们全都是相对於 identity $y = x$ 的非线性 deformation:



(10)

这和 Steven Smale 提出的「马蹄」[11] 非常类似，它是制造混沌的处方之一。换句话说，「拉扯」然后放回原空间，如此不断重复，就会产生混沌 [5] [12]。(神经网络的时间逆向就是 σ^{-1} ，所以时间向前也是混沌。) Smale 马蹄的另一个变种叫做 baker map，其作用类似於「搓面粉」。

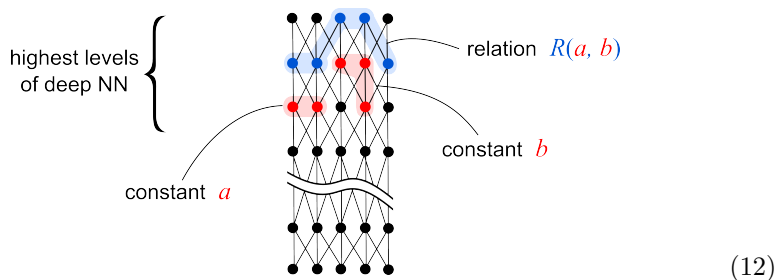
总结来说：每个神经元的输出代表某个 feature 的存在与否。
而，更高层的神经元代表下层 features 之间的关系。

凭这个思路推广，可以推测这样的 correspondence:

$$\begin{array}{lll}
 \mathcal{M} & \simeq & \mathcal{X} \\
 \text{constant} \quad \bullet & \Leftrightarrow & \text{neuron} \\
 \text{relation} \quad \bullet - \bullet & \Leftrightarrow & \text{relation between higher and lower neurons}
 \end{array} \tag{11}$$

但要注意的是这对应未必是一对一的，可能是一个 constant 对应几个 neurons 的线性组合。具体情况可能像以下的示意图（实际上每层神经网络可能有很多

神经元):



$R(a, b)$ 可以在 a, b 的 common parents 中寻找 (例如那些蓝色神经元, $R(a, b)$ 的值 = 蓝色神经元的某个线性组合)。验证的方法是: 当 a 和 b 的信号都是「有」时, $R(a, b)$ 的值也应该是 true。

4 结论

这篇论文并不太成功, 因为跳到 (11) 和 (12) 的结论没有严谨的根据, 只是直观上觉得有可能。理论上来说, 既然知道了 \mathcal{M} 那边是怎样生成的、 \mathcal{X} 那边是怎样生成的, 则要在两边建立「高速公路」应该是可行的。实际上, 似乎只要建立一个深度网络就可以, 因为神经网络是 universal function approximator, 根本不用考虑 \mathcal{M} 和 \mathcal{X} 这两个结构之间的关系。

进一步的研究, 希望数学专业的人能帮助一下:

1. 在逻辑那边, 可不可以转换成 algebraic geometry 的结构? 即是说: 逻辑式子 \simeq 代数方程。这种代数逻辑的做法, 我暂时只知道有 [2], 是很偏门的研究。
2. 能不能根据 \mathcal{M} 和 \mathcal{X} 的结构, 找出它们之间的桥的最简单形式? 可以用数学归纳法, 逐步考虑 \mathcal{M} 和 \mathcal{X} 生成的方式, 或许有帮助?

应用: 对于用深度学习做 natural language understanding 的人, 这理论或许会很有用。

5 Prior art

- Bader, Hitzler, Hölldobler and Witzel 在 2007 年提出了一个 neural-symbolic integration 的做法 [3]。他们首先由 logic theory 生成抽象的 Herbrand

model¹，再将 Herbrand model 映射到某个 fractal 空间，然后直接用神经网络学习那 fractal 空间。虽然用了 model theory，但他们没有利用到本文所说的 \mathcal{M} 和 \mathcal{X} 之间的关系。

- Khrennikov 在 1997 年开始的多篇论文中提出了用 p -adic 代数来模拟思维空间 \mathcal{X} 的结构，详见 [1] 一书。一个 p -adic 数可以看成是一个 p 进制的小数， p 是任何质数。
- 经典逻辑是二元逻辑，近代已经有无数将它扩充到 fuzzy 或 probabilistic 的尝试（作者也提出过 [15]），但仍未有统一的理论。与此不同的另一个方向，如果将点看成是 first-order objects，谓词是点空间上的函数，直接得到 metric structures 上的连续逻辑 (continuous first-order logic) [13]，这可以看成是一种 \mathcal{M} 的结构。
- 模型论中有（超滤子）ultra-filter 和 ultra-product 这些建构，它们起源於泛函分析，最近有很多横跨模型论和 Banach 空间的新研究 [7]。简单地说 ultra-product 用来将一些 models 构造出新的乘积 models。但我粗略地看过一下之后发现 ultra-product 通常涉及无穷集合，而且是很大的物体，在计算机上应用似乎不太实际。

Acknowledgement

谢谢 Ben Goertzel（OpenCog 人工智能的创始人）在 AGI mailing list 上和我的讨论。Ben 初次指出神经网络学习和逻辑 inductive 学习不同，引起我研究两者之间的关系。

References

1. Vladimir Anashin and Andrei Khrennikov. *Applied algebraic dynamics*. de Gruyter, 2009.
2. Andreka, Nemeti, and Sain. *Handbook of philosophical logic*, chapter Algebraic logic, pages 133–247. Springer, 2001.
3. Bader, Hitzler, Hödöbler, and Witzel. The core method: Connectionist model generation for first-order logic programs. *Studies in Computational Intelligence* 77, 205–232, 2007.
4. Kees Doets. *Basic model theory*. CSLI notes, 1996.
5. Robert Gilmore and Marc Lefranc. *The topology of chaos: Alice in stretch and squeezeland*. Wiley-VCH, 2011.
6. Shawn Hedman. *A first course in logic*. Oxford, 2004.
7. José Iovino. *Applications of model theory to functional analysis*. Dover, 2002.
8. Roger Maddux. *Relation algebras*. Elsevier, 2006.
9. Maria Manzano. *Model theory*. Oxford, 1999.
10. Gunther Schmidt. *Relational mathematics*. Cambridge, 2010.

¹ Herbrand model 是邏輯 AI 中常用的概念，大意是用邏輯語言 \mathcal{L} 生成「所有可以代入的東西」(instantiating whatever that can be instantiated)，由此產生的不含變量的句子 (sentence) 的集合。換句話說，Herbrand model 的特點是它只靠 \mathcal{L} 自身產生它的模型，而不依賴任何外在結構。每個邏輯 theory 都必然至少有一個 Herbrand model。

11. Stephen Smale. Differentiable dynamical systems. *Bulletin of the American Mathematical Society*, 1967.
12. Tamás Tél and Márton Gruiz. *Chaotic dynamics: an Introduction based on classical mechanics*. Cambridge, 2006.
13. Itai Ben Yaacov, Alexander Berenstein, C Ward Henson, and Alexander Usvyatsov. Model theory for metric structures. In *Model theory with applications to algebra and analysis, vol 2*. Cambridge, 2008.
14. King Yin Yan. Wandering in the labyrinth of thinking – a cognitive architecture combining reinforcement learning and deep learning. to be submitted AGI 2017.
15. King-Yin Yan. Fuzzy-probabilistic logic for common sense reasoning. *Artificial general intelligence 5th international conference, LNCS 7716*, 2012.